

The Concept of Algorithm

学习目标

- 理解算法的概念和特点
- 掌握算法的形式化表示：伪代码及流程图表示
- 运用计算思维解决实际问题及符号系统建模

0. 问题引入

地图那些事儿：路线规划

16 岁的丽丽假期到北京来旅游，乘火车到站北京南站。这是她第一次乘坐地铁，目的地第一站是 2008 年奥运会主场馆：奥林匹克公园。



图 1 北京地铁运营线路图

0.1 人们怎样解决实际问题

请自己动手帮她规划一下地铁线路。

{旁注：乘坐地铁八步曲：有序进站——接受安检——自助购票——文明刷卡——按线乘车——有序乘降——刷卡出站——文明乘梯¹}

问题建模：形式化与抽象问题

- 交通工具：地铁
- 起始站：北京南站
- 目的站：奥林匹克公园站

¹ <http://jingyan.baidu.com/article/a378c9607bb069b3282830fb.html>

- 路线规划依据：北京地铁线路图
- 求解：乘车路线

【动手实践】请学生依据上图自主设计路线。

下面列出了同学们可能设计的两种路线，也是解决问题的两种实际方案。

路线一：

- 北京南站上车（4 号线—安河桥北方向）：3 站
- 宣武门站换乘（4 号线）→（2 号线内环）：7 站
- 鼓楼大街站换乘（2 号线）→（8 号线—朱辛庄方向）：5 站
- 奥林匹克公园站下车（8 号线）

路线二：

- 北京南站上车（4 号线—安河桥北方向）：14 站
- 海淀黄庄站换乘（4 号线）→（10 号线内环）：6 站
- 北土城站换乘（10 号线）→（8 号线—朱辛庄方向）：2 站
- 奥林匹克公园站下车（8 号线）

【问题思考】上述两个路线，丽丽更有可能选择哪一个？为什么？

0.2 基于计算机的问题解决

在实际情况下，丽丽打开手机地图 App，输入起点和终点，软件自动显示不同需求下的最优路线。

【实践演示】教师演示或学生在 PC 上自己尝试地图 App

【问题思考】计算机是如何解决路线规划这个实际问题的呢？

【实践演示】人工智能聊天机器人

“机器人，帮我解决人类文明起源这个社会问题，可以吗？”

随着信息技术的发展，CPU 已经具备超过每秒 50 亿次的计算能力。人工智能已经渗透到生活的方方面面，为什么计算机不能帮我们做解决上述论文问题？

【问题思考】人如何解决问题，计算机如何解决问题？

因为计算机本身并不会思考，它们只做告诉他们的事情；也就是说，计算机只执行指令。

1. 问题解决(Problem solving)

整个第三章的目标是循序渐进的教会你如何像一个计算机科学家一样思考。这种思维方式结合了数学、自然科学、工程等等领域的最佳特点。

计算机科学家和数学家一样使用规范化语言来表示想法；和工程师一样设计事物、将各个部分组装成为系统，并评价不同选择之间的折衷；和科学家一样，观察复杂系统的行为，提出并测试假设。

对计算机科学家来说最重要的技能是问题解决，这是指用公式化（用符号系

统)表示问题、创造性的思考解决方法并清晰准确的表达解决方案。而学习编程的过程正是锻炼“问题解决”技能的绝佳机会。

同学们从小学习数学，对于怎样解数学题有着深入训练。乔治·波利亚是一位著名的数学家兼数学教育家，他的著作《怎样解题》对数学问题的解决做了深入的研究。在书中，他将解题分为弄清问题、拟定计划、实现计划和回顾四个阶段。在信息社会，我们要像计算机科学家一样，学会基于计算机的问题解决方法。下表列出了从基于数学的问题解决到基于计算机的问题解决拓展思路。

波利亚：《怎样解题》节选			基于计算机的问题解决
第一步	弄清问题	<ul style="list-style-type: none">你能用自己的语言陈述问题吗？从问题中可以获取哪些已知数据、信息和条件？未知量是什么？还缺哪些信息？	理解问题，表示为符号系统
第二步	拟定计划	<ul style="list-style-type: none">寻找是否存在特定模式回忆相关问题将问题分解为各不同的组成部分确定子目标制作表格，写出方程，猜解并验证	设计算法
第三步	执行计划	<ul style="list-style-type: none">实现步骤2中的策略，执行必要的数学计算。在执行过程中检查每一步计划对每一步工作准确记录将求解工作以直观易懂的可视化方式进行组织	编写程序
第四步	评价回顾	<ul style="list-style-type: none">检查原始问题的求解结果从原始问题出发对求解结果进行说明，你的答案合理吗？确定是否有其它答案或者解法？若可能，确定该解法适用的其它相关或者更一般性的问题	测试程序并建立可重用模块

表 1 波利亚在数学问题方面的《怎样解题》扩展到基于计算机的问题解决。

{旁注：波利亚（George Polya，1887-1985），美籍[匈牙利](#)数学家。生于[布达佩斯](#)，卒于美国。青年时期曾在布达佩斯、[维也纳](#)、巴黎等地攻读数学、物理和哲学，获博士学位。1914年在瑞士[苏黎世工业大学](#)任教，1938年任数理学院院长。1940年移居美国，历任[布朗大学](#)、[斯坦福大学](#)教授。1963年获美国数学会功勋奖。他是[法国科学院](#)、美国全国科学院和[匈牙利科学院](#)的院士。曾著有《[怎样解题](#)》、《[数学的发现](#)》、《[数学与猜想](#)》等，它们被译成多种文字，广为流传。
<http://baike.baidu.com/view/1003652.htm> }

算法设计是基于计算机问题解决的第二步，那么什么是算法呢？

2. 算法的概念

【动手实践】

五张卡牌正面朝下排成一行放在桌上。约定卡牌从小到大的顺序为：A 2 3 4 5 6 7 8 9 10 J Q K

请按照如下步骤进行操作：

- 步骤 1：从最左边的两张卡牌开始
- 步骤 2：将这两张卡牌正面朝上翻转
- 步骤 3：如果这两张牌从小到大排列：将它们再次正面朝下放置。否则，如果这两张不是升序排列：将它们交换，然后正面朝下放置。
- 步骤 4：如果这两张卡牌的右侧还有牌，请将手向右移动一张牌，再回到步骤 2。否则，将当前卡牌对的右侧那张向前取出当前卡牌序列，然后再回到步骤 1。

执行完上述步骤后的结果如何？（冒泡排序）

【动手实践】

小明有五只苹果，他打算找出一只最重的苹果送给爷爷吃，现在他手边有一台可以比较两个苹果重量的秤。请你帮小明设计方法找到最重的那个苹果。

请写出该问题的分步骤求解方法。

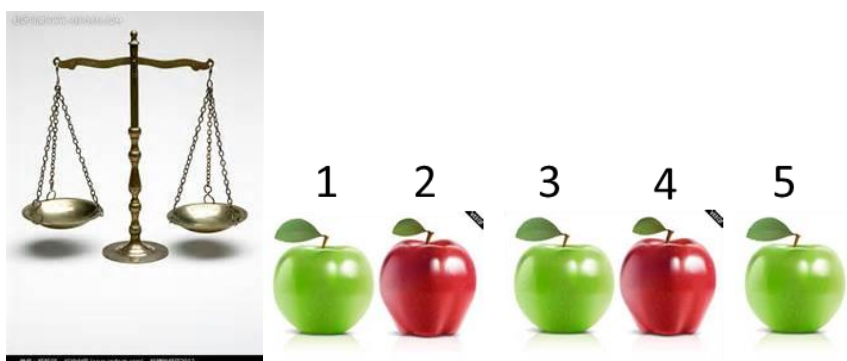


图 2 称苹果问题

问题建模：我们给苹果编上号，分别为 1 至 5 号。

已知：5 个苹果，一台只能进行重量比较的秤

求解：找到最重苹果的方法。

- 步骤 1：假设 1 号苹果是最大的，将其重量记为 A。
- 步骤 2：从碗中剩余苹果中任取一个，将其重量记为 B。
- 步骤 3：比较当前最大的苹果与刚取出苹果的重量 A 与 B。如果 B 大于 A，前往步骤 4。否则，将刚取出的苹果放置一边，继续前往步骤 2。
- 步骤 4：刚取出的苹果是当前最重的苹果，将其重量值用 A 表示。如果碗里还有苹果，前往步骤 2。
- 步骤 5：至此所有较小的苹果均已经被放置一边，保留下来的那个苹果就是最重

的，其重量以 A 表示。

1.1 算法：定义与特点

从上面的两个例子中，我们已经学会写出解决问题的分解步骤。

什么是算法？算法就是问题的分步解决方案。

算法（Algorithm）是指解题方案的准确而完整的描述，是一系列解决问题的清晰指令，算法代表着用系统的方法描述解决问题的策略机制。

<http://www.bing.com/knows/search?q=%E7%AE%97%E6%B3%95&mkt=zh-cn&FORM=BKACAI>

一个好的算法具有如下两个特点：

- 易于理解和执行
- 每一步指令清晰，没有二义性

【问题思考】

设计解决下面问题的算法。

1. 石老师收集各种玉石，他想把玉石藏品按照重量进行排列。他只有一个能比较两个物体谁更重的称重秤。
 - 请设计将 3 块重量未知玉石按重量进行排列的算法。
 - 如果是 4 块玉石呢，该如何设计排列算法？
 - 当石老师将 4 块玉石按照重量排列好后，他突然发现还有第 5 块玉石需要加入收藏。请设计一种最有效的方法帮助石老师确定应该这第 5 块玉石应该放在什么位置？

2. 农夫需要将一匹狼、一只山羊和一颗白菜运送过河。农夫的船非常小，一次只能运送一名乘客（白菜也算一名乘客）。如果农夫将狼和羊单独留在一起，狼会吃了羊。如果将羊和白菜单独留在一起，羊会吃了白菜。请设计算法帮助农夫安全将狼、羊和白菜都安全运送过河。

参考解法如下：

- 带羊过河，空船回来
- 带狼过河，带羊回来
- 带白菜过河，空船回来
- 带羊过河

要设计解决问题的合理有效算法，我们需要建立培养自己的计算思维。

3. 计算思维

计算思维（Computational Thinking）概念的提出是计算机学科发展的自然产物。计算思维是运用计算机科学的基础概念去求解问题、设计系统和理解人类的行为；计算思维最根本的内容，即其本质是抽象和自动化。在对计算思维进行了长达 5 年的跟踪研究和教学实践的基础上，教育部高教司在 2012 年设立了以计算思维为切入点的“大学计算机课程改革项目”。

<http://msra.cn/zh-cn/connections/course/comthinking/default.aspx>

【计算机科学家】

Seymour Papert 是 Logo 编程语言的发明者之一，并致力于帮助学生建立计算思维。

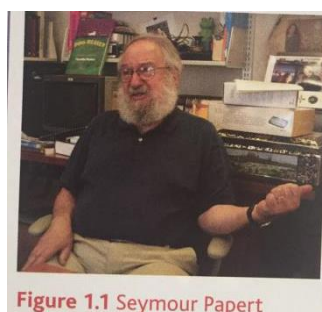


图 3 计算机科学家：Seymour Papert

周以真，现任微软研究院副院长，在她加入微软之前，曾任卡耐基梅隆大学计算机系教授，并致力于推进计算思维教育，卡耐基梅隆大学至今仍设有计算思维中心。她认为计算思维是孩子必须掌握的一项基本能力，以帮助解决生活中各种各样的实际问题。



图 4 计算机科学家：周以真

计算思维有如下特点：抽象思考、超前思考、过程思考、逻辑思考、并行思考、递归思考、模块化思考等等。

下面以模块化思考为例，说明其在问题解决算法设计步骤的具体应用。

【动手实践】请设计算法画出如下图案。

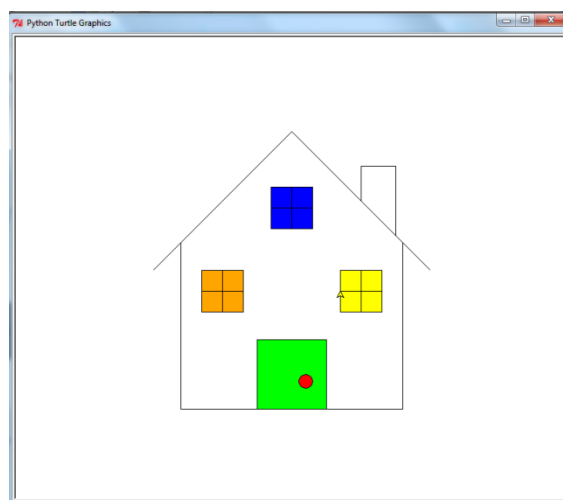


图 5 画房子图案

这里我们要介绍模块化的思想, 请将下列要画的各个要素填入下图: 烟囱、轮廓、门把手、门框、门、窗户。

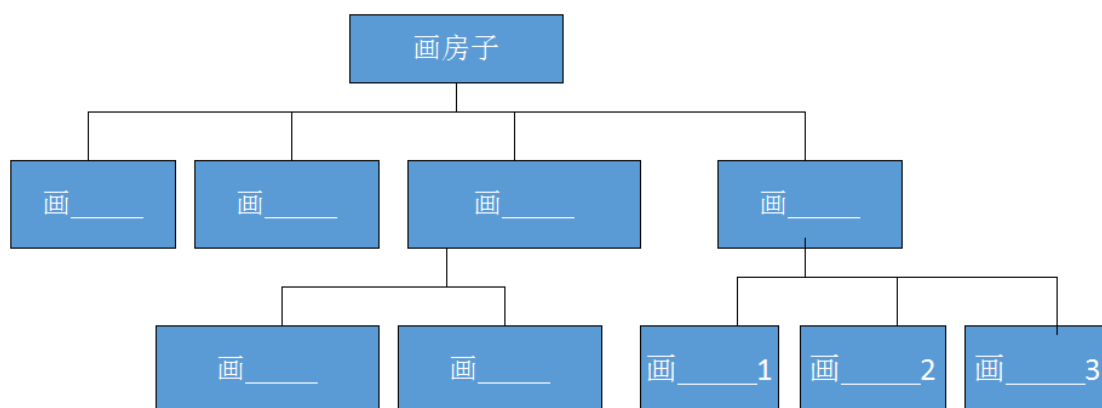


图 6: 分解画房子问题

【动手实践】运行所附代码 `DrawHouse.py`, 观察程序结构。请用自己的话描述什么是模块化思想? 能举出其它具体的应用实例吗?

模块化是指解决一个复杂问题时自顶向下逐层把软件系统划分成若干模块的过程。每个模块完成一个特定的子功能, 所有的模块按某种方法组装起来, 成为一个整体, 完成整个系统所要求的功能。<http://wenda.chinabaike.com/html/20101/q1104581.html>

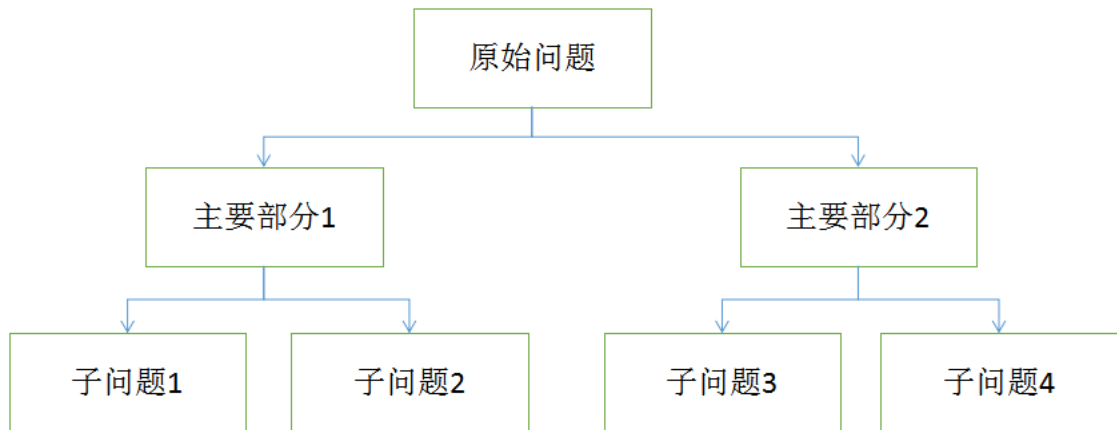


图 7： 自顶向下的模块化思想 1

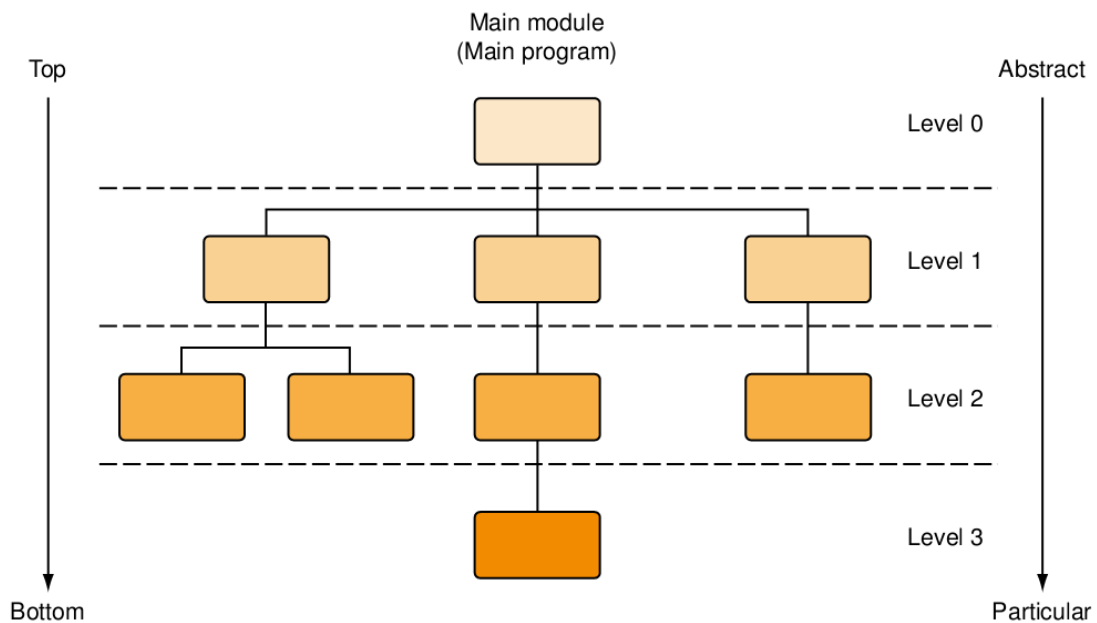


图 8： 自顶向下的模块化思想 2

综上所述，模块化的核心思想是：将大问题逐渐分解为可以解决的小问题。

下面我们来看基于计算机的问题解决第三步：编写程序。

4. 程序与程序设计

打个比方，算法是计划，程序是具体实现。

程序是用来指定如何进行计算的指令序列。计算可以是数学的，例如求解方程组、求解多项式方程的根，但也可能是符号计算，例如搜索或者替换文档中的文本等。不同程序设计语言的细节有所不同，但是它们都有一些基本指令：

输入：从键盘、文件或者其它输入设备获取数据。

- 输出：将数据显示在屏幕上或者将数据发送给文件或者其他输出设备。
- 处理/计算：基本的数学运算，如加、减、乘、除
- 条件执行/选择结构：检查某些条件成立与否，从而执行对应的语句
- 重复/循环：重复执行某些指令，但每次执行通常有一些变化。

上面就是算法的基本组成部分，无论程序多么的复杂，其都是由上面的基本部分组成。因此，我们可以将算法与程序设计描述为将大型复杂任务分解为越来越小的子任务，直到当前子任务足够简单，从而能够用上述基本结构进行解决。

计算机需要通过特殊的语言来给其指令，本书中我们将使用 Python 这种高层编程语言。Logo 编程语言由 Seymour Papert 等人发明，其目的是帮助儿童学习编程及数学概念。Logo 非常容易学，其最著名的功能是海龟画图，一个叫海龟的屏幕对象能够根据指令在屏幕上画出各种图案。如今 Logo 语言在演示计算思维中如抽象和分解等主题时仍然非常有用。海龟画图也可以通过 Python 实现，只要 Python 的 turtle 库被加载到编程环境中，海龟画图的所有功能就都可以使用了。

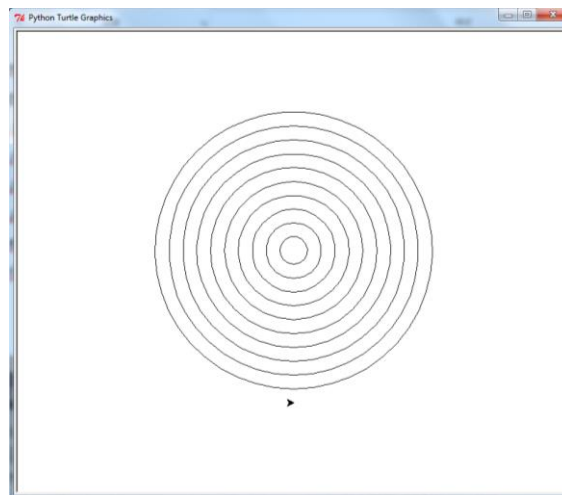


图 9 画图算法实例

【动手实践】请设计算法使得计算机能够自动画出图 9 图案。

下面是 Python turtle 画图的一个简单演示程序，其运行结果就是图 9。

```
import turtle
NumCircles = 10
CircleInterval = 20
turtle.home()
for i in range(1,1+NumCircles):
    turtle.pendown()
    turtle.circle(CircleInterval*i)
    turtle.penup()
```

```
turtle.goto(0,-1*CircleInterval*i)
turtle.home()
turtle.mainloop()
```

请运行所附代码：DrawCircles.py，思考算法与程序之间的关系。

5. 算法的描述方法

算法有两种常用的描述方法：流程图与伪代码。

5.1 流程图

流程图是一种图形化的算法表示方法，常用的流程图基本符号，其表示含义及具体事例如表 2 所示。流程图在实际问题中的应用请参见第 6 小节：算法设计实例。








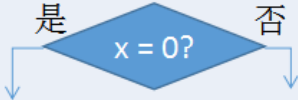

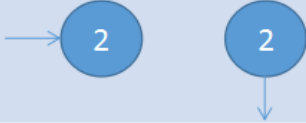
符号	意义	示例
	起止框	
	输入输出框	
	计算或者处理框	
	判断框	
	连接点	

表 2： 流程图基本符号、含义及实例。

5.2 伪代码

伪码（Pseudocode）是一种算法描述语言。使用伪码的目的是使被描述的算法可以容易地以任何一种编程语言（Pascal，C，Java 等）实现。因此，伪代码必须结构清晰、代码简单、可读性好，并且类似自然语言。 介于自然语言与编程语言之间。使用伪代码， 不用拘泥于具体实现。相比程序语言（例如 Java，C++，C，Dephi 等等）它更类似自然语言。

<http://www.bing.com/knows/search?q=%E4%BC%AA%E4%BB%A3%E7%A0%81&mkt=zh-cn&FORM=BKACAI>

伪代码由关键词与所需变量(值可能发生变化)/常量（值不变）组成。

关键词	关键词伪代码表示
输入	INPUT
输出	OUTPUT
赋值	←
判断	IF <条件> THEN <语句序列> ENDIF
循环	WHILE <条件> <语句序列>

表 3： 伪代码主要结构及对应关键词



图 10 循环结构与条件选择结构的不同实现及其伪代码表示
伪代码在实际问题中的应用请参见第 6 小节：算法设计实例。

6. 算法设计与表示实例

本小节以猜数字游戏为例，对算法设计、计算思维及算法的两种描述方式进行实际应用。

很多人都玩过这个游戏：甲在心中想好一个数字，乙来猜。每猜一个数字，甲必须告诉他是猜大了，猜小了，还是刚好猜中了。

我们来模拟这个游戏过程，其中用户充当甲的角色，计算机充当乙的角色。假设猜数范围在 1-100 内的整数。为了更快地猜中，请为计算机设计算法，并分别用流程图和伪代码表示。

【思考】请同学们思考如何为计算机设计猜数算法。

当然，随机猜数是一种选择，但是当猜数范围较大时，如在 0-1000 以内猜一个数，随机猜所需要的次数可能很多。因此，我们考虑二分法，也即每次猜中间值，如果所猜中间值比答案要大，那么我们接下来只需要看中间值前面的那一半数，反之如果所猜中间值比答案要小，那么接下来只需要看中间值后面的那一半数。从而，每猜一次我们可以将范围缩小一半。那么二分法的算法就比随机猜要高效很多。

下面举一个具体例子对算法进行详细说明：以 0-100 范围内猜数字，待猜数字为 30 为例。

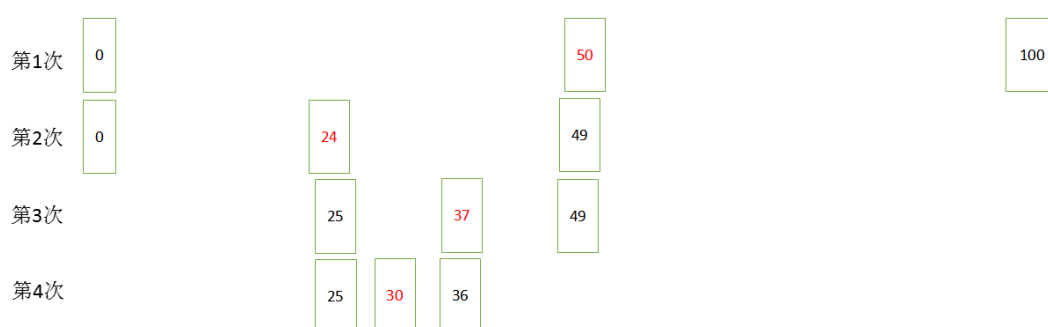


图 11 0-100 范围内猜数字，待猜数字为 30 的算法过程示意图。

具体算法过程如下：

- 第 1 次猜数字为： $(0+100) / 2 = 50$ 。反馈为所猜数字偏大，因此猜数范围缩小为 0-49。
- 第 2 次猜数字为： $(0+49) / 2 = 24$ 。反馈为所猜数字偏小，因此猜数范围缩小为 25-49。
- 第 3 次猜数字为： $(25+49) / 2 = 37$ 。反馈为所猜数字偏大，因此猜数范围缩小为 25-36。
- 第 4 次猜数字为： $(25+36) / 2 = 30$ 。计算机猜对了！

接下来，我们分别用伪代码和流程图的方法对猜数字游戏的具体算法进行描述。

猜数字游戏：伪代码算法描述

```
low←0    #猜数范围下限变量
high←100  #猜数范围上限变量
i←1      #猜测次数计数变量
WHILE TRUE: #循环结构，条件为真，只有当遇到 BREAK 语句才会退出循环
    MID ← (low+high)/2  # 取猜数范围中间值作为当前猜数值
    OUTPUT 计算机第 i 次猜测数是：MID  # 屏幕显示输出告知用户当前计算机的猜测值
```

```

    OUTPUT 请告诉计算机这个数猜对了：y，猜大了：b，猜小了：s
#屏幕显示告知用户如何给计算机反馈信息
    INPUT 用户输入（是 y, b, s 这三个字母中的一个）→ Feedback #从键盘获取用户
反馈信息
    IF Feedback = 'y'      #如果用户输入 'y'，说明计算机猜对了
        OUTPUT “恭喜你猜对了！”
        BREAK              #程序结束
    ENDIF
    IF Feedback = 'b'
        #如果用户输入 'b'，说明计算机猜大了，那么无需再考虑比当前猜测值更大的
        那些值
        THEN
            high ← MID -1   #将猜数范围上限修改为当前猜测值-1
        ELSE
            #如果用户输入 's'，说明计算机猜大了，那么无需再考虑比当前猜测值更小的那
            些值
            THEN
                low ← MID +1   #将猜数范围下限修改为当前猜测值+1
            ENDIF
        i ← i + 1           #猜测次数增加 1
    END WHILE

```

请注意：← 表示赋值符号，= 表示判断等号两边是否相等，#符号后面的内容表示注释，也即对本行伪代码的具体解释说明。这段伪代码里面有两种算法结构：1) WHILE END WHILE 的循环结构，2) IF ... THEN ... ELSE THEN ... END IF 的条件选择结构。

猜数字游戏：流程图算法描述

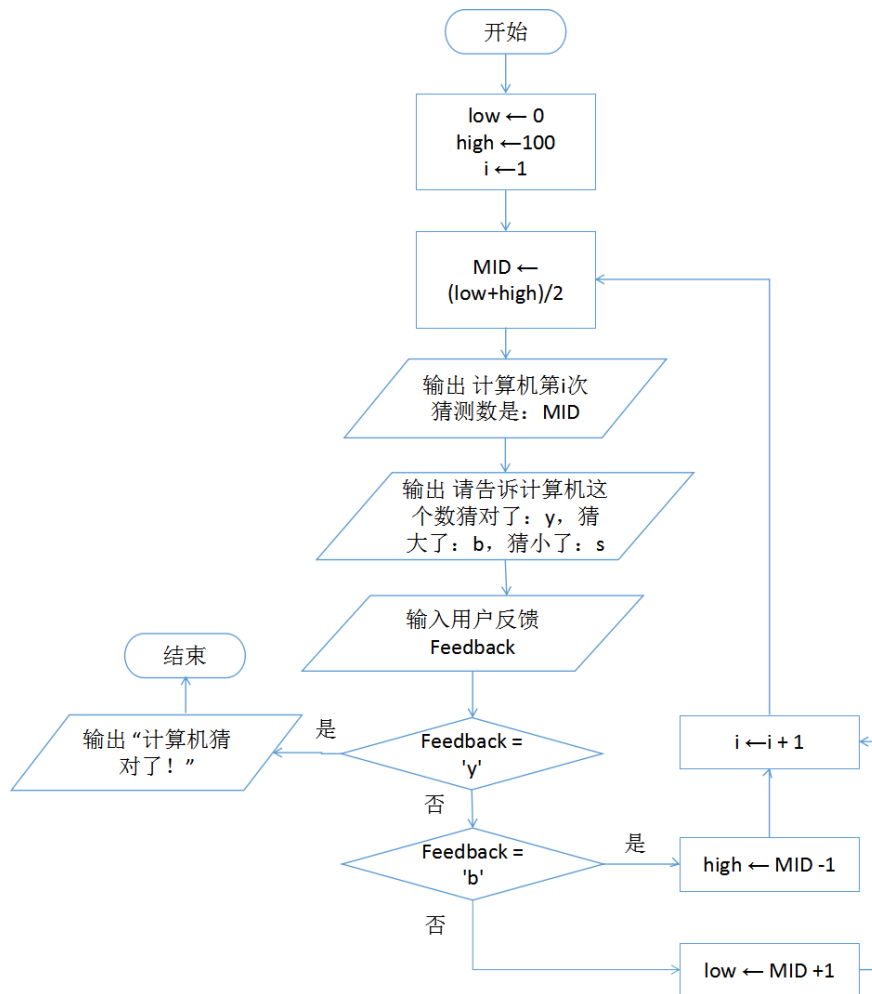


图 12：猜数字游戏：流程图算法描述

注：流程图右侧从下往上的箭头表示循环结构。

【动手实践】请运行所附代码 `Guessnumber.py`，并对照伪代码与流程图对算法进行检验与思考。

【练习题】

请对如下问题设计算法并分别用伪代码和流程图表示

1. 摄氏度与华氏度之间的互相转换。

已知：摄氏温度 c ，将其转化为华氏温度 f ，转换公式为： $f=c*9/5+32$

2. 成绩等级计算, 100 分制

$100 \geq \text{成绩} \geq 90$ 等级 A*

$90 > \text{成绩} \geq 80$ 等级 A

$80 > \text{成绩} \geq 70$ 等级 B

$70 > \text{成绩} \geq 60$ 等级 C

$60 > \text{成绩} \geq 0$ 等级 D

参考文献:

- OCR computer science book
- Computer science illuminated
- How to think like a computer scientist: open book
- Wikipedia